SQLite Queries and Python GUI

Connor Baldwin
Duy Tran
Esteban Reynaga

HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.
I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code

Task 1

Q1

```
alter table book_loans ADD COLUMN Late INTEGER;
update book_loans
set Late = case
  when (julianday(returned_date) > julianday(due_date)) then 1
  else 0
end;
```

book_id	branch_id	card_no	date_out	due_date	Returned_date	Late
1	1	123456	2022-01-01	2022-02-01	2022-02-01	0
2	1	789012	2022-01-02	2022-02-02	NULL	0
3	2	345678	2022-01-03	2022-02-03	NULL	0
4	3	901234	2022-01-04	2022-02-04	2022-02-04	0
5	1	567890	2022-01-05	2022-02-05	2022-02-09	1
6	2	234567	2022-01-06	2022-02-06	2022-02-10	1
7	2	890123	2022-01-07	2022-02-07	2022-03-08	1
8	3	456789	2022-01-08	2022-02-08	2022-03-10	1
9	1	111111	2022-01-09	2022-02-09	2022-02-06	0
10	2	222222	2022-01-10	2022-02-10	2022-02-07	0
11	1	333333	2022-03-01	2022-03-08	2022-03-08	0
12	3	444444	2022-03-03	2022-03-10	2022-03-10	0
13	3	555555	2022-02-03	2022-03-03	2022-02-18	0
14	1	565656	2022-01-14	2022-02-14	2022-03-31	1
15	3	676767	2022-01-15	2022-02-15	2022-02-21	1
16	2	787878	2022-03-05	2022-03-12	2022-03-24	1
17	3	989898	2022-03-23	2022-03-30	2022-03-30	0
18	3	121212	2022-01-18	2022-02-18	2022-02-18	0
19	1	232323	2022-03-24	2022-03-31	2022-03-31	0
20	3	343434	2022-01-21	2022-02-21	2022-02-21	0
21	3	454545	2022-01-24	2022-02-24	2022-02-24	0

Action output response: 21

Q2

alter table library_branch ADD COLUMN LateFee double;
update library_branch
set LateFee = branch_id + 0.99

branch_id	branch_name	branch_address	LateFee
1	Main Branch	123 Main St, New York, NY 10003	1.99
2	West Branch	456 West St, Arizona, AR 70622	2.99
3	East Branch	789 East St, New Jersy, NY 32032	3.99
4	North Branch	456 NW, Irving, TX 76100	4.99
5	UTA Branch	123 Cooper St, Arlington TX 76101	5.99

Action output response: 5

Q3

```
create view vBookLoanInfo as
        select
        BL.card_no as Card_no,
        BO.name as "Borrower Name",
        BL.date_out as Date_Out,
        BL.due_date as Due_Date,
        BL.Returned_date as Returned_date,

        case
        when BL.Returned_date is not null then
                julianday(BL.Returned_date) - julianday(BL.date_out)
        else null
        end as TotalDays,

        B.title as "Book Title",

        case
        when BL.Returned_date is null then null
        when julianday(BL.Returned_date) > julianday(BL.due_date) then
                julianday(BL.Returned_date) - julianday(BL.due_date)
        else 0
        end as "Num of days returned late",

        BL.branch_id as "Branch ID",

        case
        when BL.Returned_date is null then null
```

```
when julianday(BL.Returned_date) > julianday(BL.due_date) then
        (julianday(BL.Returned_date) - julianday(BL.due_date)) * LB.LateFee
else 0
end as LateFeeBalance
from book_loans BL
join borrower BO on BL.card_no = BO.card_no
join book B on BL.book_id = B.book_id
join library_branch LB on LB.branch_id = BL.branch_id;
```

Select * from vBookLoanInfo;

Card_no	Borrower Name	Date_Out	Due_Date	Returned_date	TotalDays	Book Title	Num of days returned late	Branch ID	LateFeeBalance
123456	John Smith	2022-01-01	2022-02-01	2022-02-01	31.0	To Kill a Mockingbird	0	1	0
789012	Jane Doe	2022-01-02	2022-02-02	NULL		1984	0	1	0
345678	Bob Johnson	2022-01-03	2022-02-03	NULL		Pride and Prejudice	0	2	0
901234	Sarah Kim	2022-01-04	2022-02-04	2022-02-04	31.0	The Great Gatsby	0	3	0
567890	Tom Lee	2022-01-05	2022-02-05	2022-02-09	35.0	One Hundred Years of Solitude	4.0	1	7.96
234567	Emily Lee	2022-01-06	2022-02-06	2022-02-10	35.0	Animal Farm	4.0	2	11.96
890123	Michael Park	2022-01-07	2022-02-07	2022-03-08	60.0	The Catcher in the Rye	29.0	2	86.71
456789	Laura Chen	2022-01-08	2022-02-08	2022-03-10	61.0	Lord of the Flies	30.0	3	119.7
111111	Alex Kim	2022-01-09	2022-02-09	2022-02-06	28.0	Brave New World	0	1	0
222222	Rachel Lee	2022-01-10	2022-02-10	2022-02-07	28.0	The Picture of Dorian Gray	0	2	0
333333	William Johnson	2022-03-01	2022-03-08	2022-03-08	7.0	The Alchemist	0	1	0
444444	Ethan Martinez	2022-03-03	2022-03-10	2022-03-10	7.0	The God of Small Things	0	3	0
555555	Grace Hernandez	2022-02-03	2022-03-03	2022-02-18	15.0	Wuthering Heights	0	3	0
565656	Sophia Park	2022-01-14	2022-02-14	2022-03-31	76.0	The Hobbit	45.0	1	89.55
676767	Olivia Lee	2022-01-15	2022-02-15	2022-02-21	37.0	The Lord of the Rings	6.0	3	23.94
787878	Noah Thompson	2022-03-05	2022-03-12	2022-03-24	19.0	The Hitchhiker's Guide to the Galaxy	12.0	2	35.88
989898	Olivia Smith	2022-03-23	2022-03-30	2022-03-30	7.0	The Diary of a Young Girl	0	3	0
121212	Chloe Park	2022-01-18	2022-02-18	2022-02-18	31.0	The Da Vinci Code	0	3	0
232323	William Chen	2022-03-24	2022-03-31	2022-03-31	7.0	The Adventures of Huckleberry Finn	0	1	0
343434	Olivia Johnson	2022-01-21	2022-02-21	2022-02-21	31.0	The Adventures of Tom Sawyer	0	3	0
454545	Dylan Kim	2022-01-24	2022-02-24	2022-02-24	31.0	A Tale of Two Cities	0	3	0

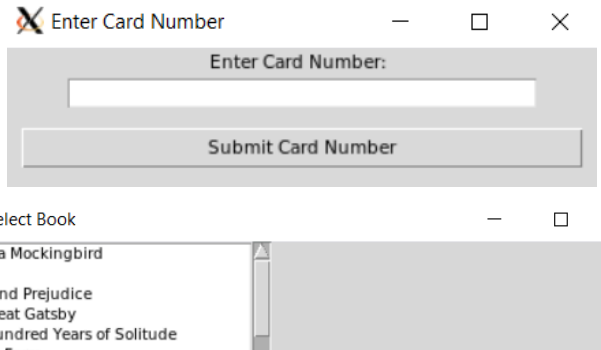
Action output response: 21 Rows

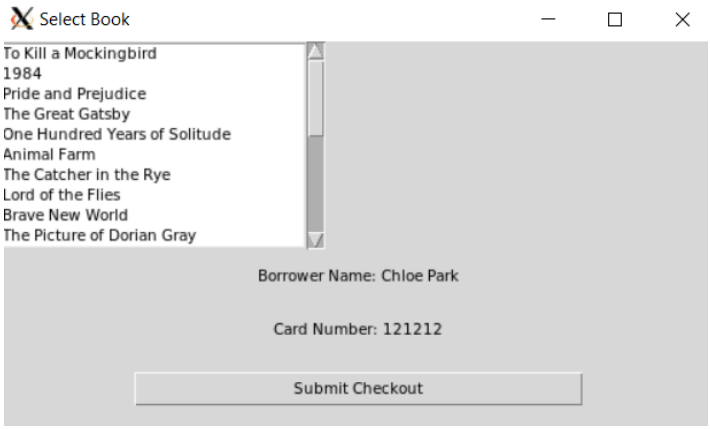
Task 2: GUI Requirements

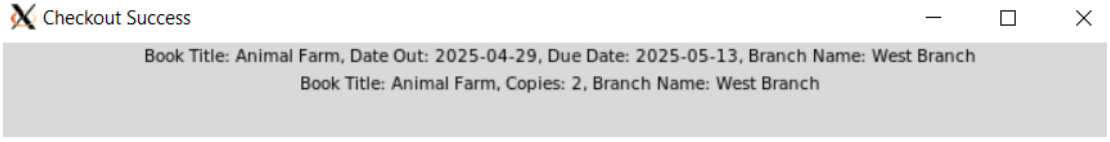
The Main Window where you can access each requirement's flow

Requirement 1

After pressing "Add Borrower" the user must input a valid card number to open another window and select a book, ending with a display of the book loan information as well as the book information

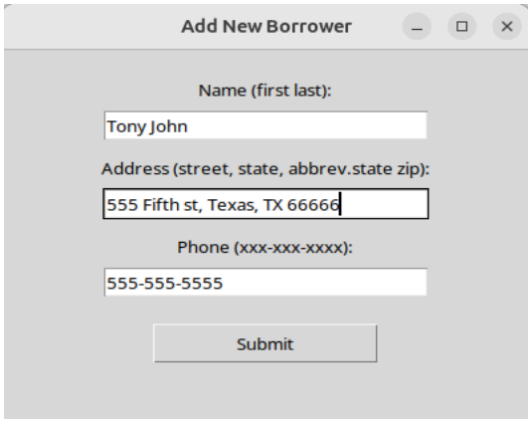


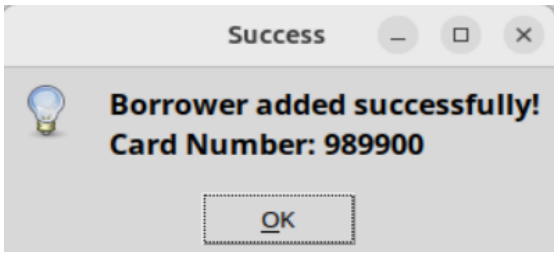




2

If a person wanted to add a new borrower, he would press the “Add Borrower” button in the red section. Once the button is pressed, a new window will appear. In this new window it will present 3 fields the person must fill out in the correct format, name, address, and phone number. After filling out the necessities, the person will need to press the “Submit” button at which a card number will appear that will be assigned to that person. Then click “ok” closing all windows.



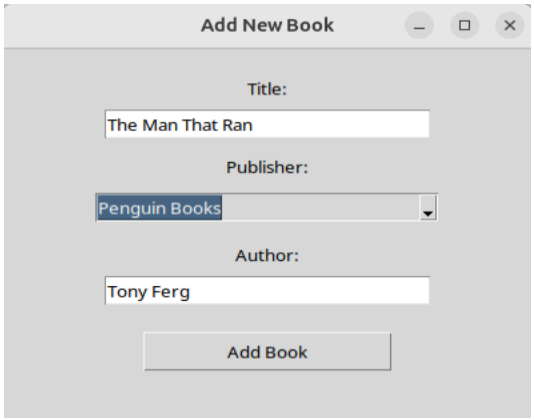


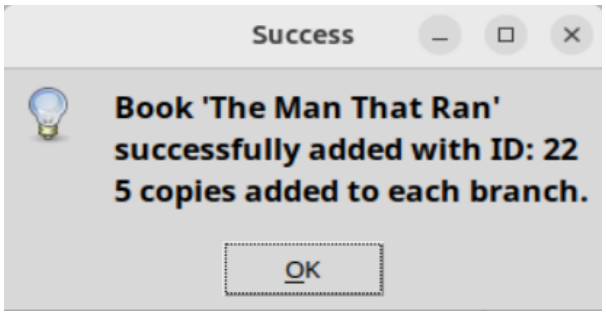
Editable SQL query:

```
        borrowerCur.execute("""
                INSERT INTO borrower (name, address, phone)
                VALUES (?, ?, ?)
        """, (name, address, phone))

        borrowerCur.execute("SELECT card_no FROM borrower ORDER BY card_no DESC
LIMIT 1")
```

3

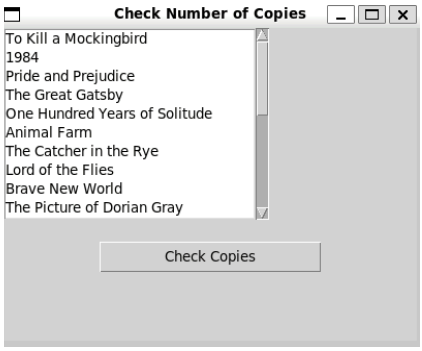
Press “Add New Book” to add a new book with publisher and author name, adding 5 copies to each branch. New window will pop up. In the new window type in book title, select publisher from dropdown, and type author’s name. Press “Add Book” to confirm inputs in fields. Another new window will have a message stating confirmation.

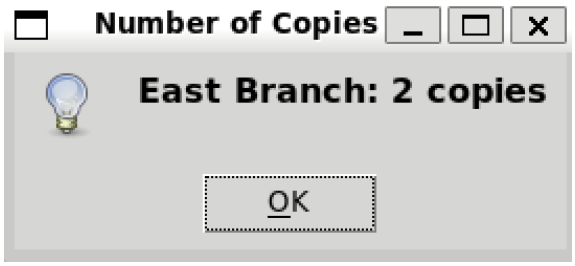




4

Upon selecting "Check Number of Copies," a new window will appear, displaying a list of available books across all libraries. Users can then select a book from this list and click "Check Copies" to view the number of copies at each branch where it is available.



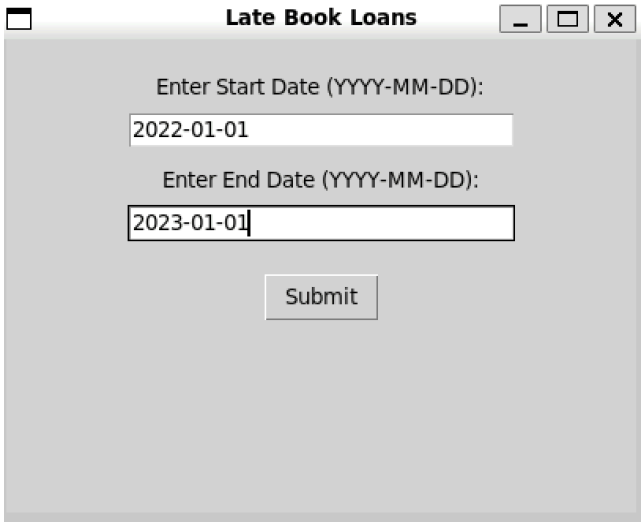


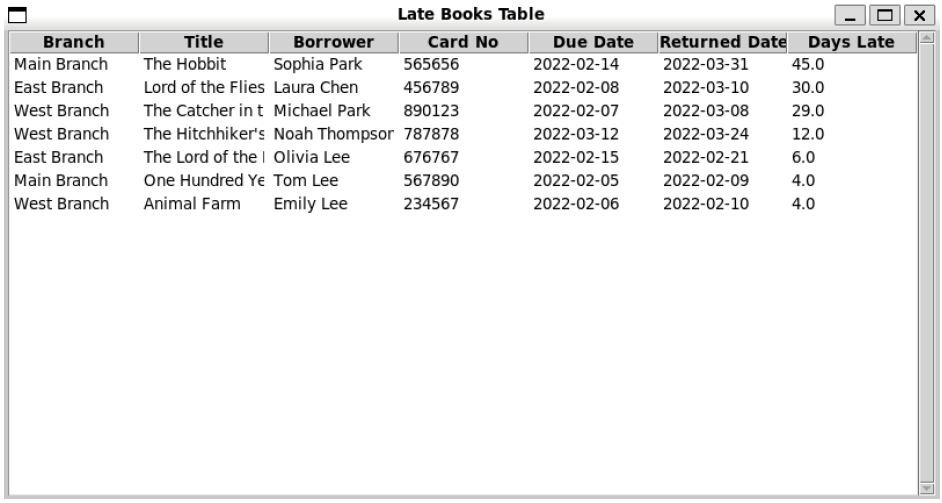
Editable SQL Query:

```
local_cur.execute("""
                    select lb.branch_name, bc.no_of_copies, b.book_id
                    from BOOK B
                    natural join book_copies bc
                    natural join library_branch lb
                    where b.title = ?
            """, (selectedBookTitle.get(),))

cur.execute("select b.title from book B")

check_cur.execute("""
                    select lb.branch_name, bc.no_of_copies
                    from book b
                    natural join book_copies bc
                    natural join library_branch lb
                    where b.title = ?
            """, (title,))
```

5

After clicking "Late Book Loans," a new window will open. This window contains two text fields for the user to input a start and end date. These dates define the period for identifying users with overdue book loans. Once the desired start and end dates are entered and the user clicks "Submit," a new window will appear. This window will display a table listing borrowers who had late book loans within the specified date range.





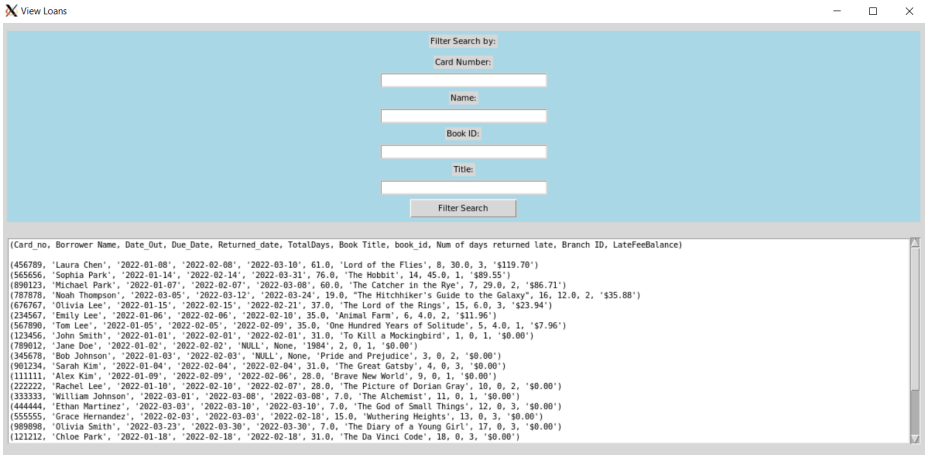
Editable SQL Query:

```
cur.execute("""
        select
        lb.branch_name,
        b.title,
        bo.name,
        bl.card_no,
        bl.due_date,
```

```
        bl.Returned_date,
        julianday(bl.Returned_date) - julianday(bl.due_date) as days_late
        from book_loans bl
        natural join book b
        join library_branch lb on bl.branch_id = lb.branch_id
        join borrower bo on bl.card_no = bo.card_no
        where bl.due_date BETWEEN ? and ?
        and bl.Returned_date is not  "NULL"
        and bl.Returned_date > bl.due_date
        order by days_late desc
""", (startDate, endDate))
```

6

After pressing “View Loans” the user will be shown a new window where they can see all the current loans, as well as filter by entering information into the four text boxes.



Editable SQL Query:

```
loanCur.execute("ALTER TABLE book_loans ADD COLUMN Late INTEGER")

loanCur.execute("""
        UPDATE book_loans
        SET Late = CASE
        WHEN (julianday(returned_date) > julianday(due_date)) THEN 1
        ELSE 0
        END
""")

loanCur.execute("ALTER TABLE library_branch ADD COLUMN LateFee double")
```

```
loanCur.execute("UPDATE library_branch SET LateFee = branch_id + 0.99")

loanCur.execute("DROP VIEW IF EXISTS vBookLoanInfo")

loanCur.execute("""
        CREATE VIEW vBookLoanInfo AS
        SELECT
        BL.card_no as Card_no,
        BO.name as "Borrower Name",
        BL.date_out as Date_Out,
        BL.due_date as Due_Date,
        BL.Returned_date as Returned_date,

        CASE
        WHEN BL.Returned_date IS NOT NULL THEN
                julianday(BL.Returned_date) - julianday(BL.date_out)
        ELSE NULL
        END as TotalDays,

        B.title as "Book Title",
        B.book_id as book_id,

        CASE
        WHEN BL.Returned_date IS NULL THEN NULL
        WHEN julianday(BL.Returned_date) > julianday(BL.due_date) THEN
                julianday(BL.Returned_date) - julianday(BL.due_date)
        ELSE 0
        END as "Num of days returned late",

        BL.branch_id as "Branch ID",

        CASE
        WHEN BL.Returned_date IS NULL THEN NULL
        WHEN julianday(BL.Returned_date) > julianday(BL.due_date) THEN
                (julianday(BL.Returned_date) - julianday(BL.due_date)) * LB.LateFee
        ELSE 0
        END as LateFeeBalance
        FROM book_loans BL
        JOIN borrower BO on BL.card_no = BO.card_no
        JOIN book B on BL.book_id = B.book_id
        JOIN library_branch LB on LB.branch_id = BL.branch_id
""")

loanCur.execute(query, params)
```

Contribution List

Connor Baldwin
- Q1, Q2
- Python Code
    - 1, 6b

Duy Tran
- Q2
- Python Code
    - 2, 3, 6a

Esteban Reynaga
- Q3
- Python Code
    - 4, 5